TEMA 3: Regresión (segunda parte)

Regresión con datos de alta dimensión

José R. Berrendero

Departamento de Matemáticas, Universidad Autónoma de Madrid

Introducción

Si el número de variables regresoras d es muy grande, el método de mínimos cuadrados pierde algunas de sus buenas propiedades.

- Si d está cerca de n la varianza de los estimadores puede crecer dramáticamente debido al sobreajuste.
- Si hay muchas variables regresoras es verosímil que muchas de ellas no sean relevantes para predecir el valor de la variable respuesta.
- Es también probable que haya variables redundantes muy correladas entre sí.

Matemáticamente, la matriz $X'X$ estará *mal condicionada*, lo que significa que está cerca de ser singular.

Ejemplo: efecto de multicolinealidad

```
set.seed(12)
n <- 100
rho <- 0.995
Sigma <- matrix(c(1, rho, rho, 1), 2)
x <- mvrnorm(n, mu = c(0,0), Sigma = Sigma)
x1 <- x[,1]
x2 <- x[,2]

y <- x1 + x2 + rnorm(n, sd=4)
datos <- data.frame(y, x1, x2)
summary(lm(y ~ x1 + x2, data = datos))
```

Ejemplo

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2090  -2.4977  -0.1036   2.8600   7.2358
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1006     0.4037  -0.249    0.804
## x1            4.3641     4.0692   1.072    0.286
## x2           -1.8250     4.0618  -0.449    0.654
##
## Residual standard error: 4.034 on 97 degrees of freedom
## Multiple R-squared:  0.2346,    Adjusted R-squared:  0.2188
## F-statistic: 14.86 on 2 and 97 DF,  p-value: 2.34e-06
```

Tres tipos de soluciones

- **Selección de variables:** identificar un subconjunto pequeño de variables relevantes y aplicar mínimos cuadrados únicamente a este subconjunto.
- **Reducción de la dimensión:** Se proyectan las variables regresoras en un número pequeño de direcciones y se aplica el método de mínimos cuadrados a las proyecciones.
- **Regularización:** Se restringen los valores de los estimadores para reducir la varianza a cambio de introducir un sesgo, con un efecto final positivo sobre el error cuadrático medio.

Tanto la reducción de la dimensión como la selección de variables se pueden entender como casos particulares de regularización.

Selección de variables: método exhaustivo

Para seleccionar el mejor modelo entre d variables regresoras.

- Sea M_0 el modelo sin ninguna variable, que únicamente usa la media de las respuestas para predecir.
- Para $k = 1, 2, \ldots, d$,
  - Ajustar todos los modelos con k variables regresoras.
  - Seleccionar el mejor de estos modelos (el que da un valor de R^2 más alto). Sea M_k este modelo óptimo con k variables.
- Seleccionar, mediante algún criterio adecuado, el mejor modelo entre $M_0, M_1, \ldots, M_d$.

El número de modelos a considerar crece rápidamente con d. Si $d = 20$, ya hay más de un millón de modelos posibles a considerar.

Este método exhaustivo no es factible en problemas de dimensión moderada o grande.

Paso a paso hacia adelante

- Sea M_0 el modelo sin ninguna variable, que únicamente usa la media de las respuestas para predecir.
- Para $k = 0, 1, 2, \ldots, d-1$,
  - Ajustar los $d - k$ modelos que resultan de incrementar M_k con una única variable regresora.
  - Seleccionar el mejor de estos modelos (el que da un valor de R^2 más alto). Sea M_{k+1} este modelo óptimo.
- Seleccionar, mediante algún criterio adecuado, el mejor modelo entre $M_0, M_1, \ldots, M_d$.

Usando este método solo es necesario ajustar $1 + \sum_{k=0}^{d-1}(d-k) = 1 + d(d+1)/2$ modelos. Pero no garantiza la selección del mejor modelo global.

Se puede aplicar incluso en el caso en que $n < d$.

Criterios de selección de modelos

La suma de cuadrados de los residuos y el coeficiente de determinación no pueden usarse para comparar modelos con diferente número de variables. A medida que añadimos variables a un modelo, SCR siempre disminuye y R^2 siempre aumenta.

Se han desarrollado distintos criterios para seleccionar modelos:

Criterio de Mallows

Para modelos ajustados por mínimos cuadrados con p variables regresoras, se define

$$C_p = \frac{1}{n}(\mathrm{SCR} + 2p\hat{\sigma}^2),$$

donde $\hat{\sigma}^2$ es un estimador insesgado de la varianza de los errores ϵ del modelo.

Criterios de selección de modelos

Criterio de información de Akaike (AIC)

En el caso de regresión bajo la hipótesis de normalidad, mínimos cuadrados y máxima verosimilitud coinciden y el criterio AIC se reduce a:

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}(\mathrm{SCR} + 2p\hat{\sigma}^2).$$

En modelos de regresión lineal el criterio de Mallows y el AIC son proporcionales.

Criterio de información bayesiano (BIC)

Salvo constantes no relevantes se reduce a:

$$\mathrm{BIC} = \frac{1}{n}(\mathrm{SCR} + \log(n)p\hat{\sigma}^2).$$

Para los tamaños muestrales usuales, BIC penaliza más que C_p los modelos con muchas variables, por lo que suele seleccionar modelos más sencillos.

Criterios de selección de modelos

Coeficiente de determinación ajustado

Se trata de una modificación del coeficiente de determinación, para penalizar los modelos con muchas variables:

$$R^2_a = 1 - \frac{\mathrm{SCR}/(n-p-1)}{\mathrm{SCT}/(n-1)} = 1 - \frac{S^2_R}{S^2_y}.$$

Al añadir una nueva variable, R^2_a solo aumenta si se reduce la varianza residual.

Criterios de selección de modelos

Validación cruzada

Alternativamente a los criterios anteriores, siempre es posible utilizar alguna modalidad de validación cruzada para comparar.

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_{(-i)})^2 = \sum_{i=1}^{n}\left(\frac{Y_i - \hat{Y}_i}{1-h_{ii}}\right)^2$$

Sustituimos h_{ii} por su valor promedio, que es p/n, y usamos

$$\frac{1}{(1-x)^2} \approx (1+x)^2 \approx 1+2x, \quad \text{para } \; x \approx 0$$

Entonces, si $p << n$,

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_{(-i)})^2 \approx \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2\left(1+\frac{2p}{n}\right) = \text{SCR} + 2p\hat{\sigma}$$

donde $\hat{\sigma}^2 = \text{SCR}/n$

Ejemplo

Conjunto de datos simulados $n = 50$. El modelo es

$$Y_i = x_{i,1} + x_{i,2} + \epsilon_i, \;\; i = 1, \ldots, n.$$

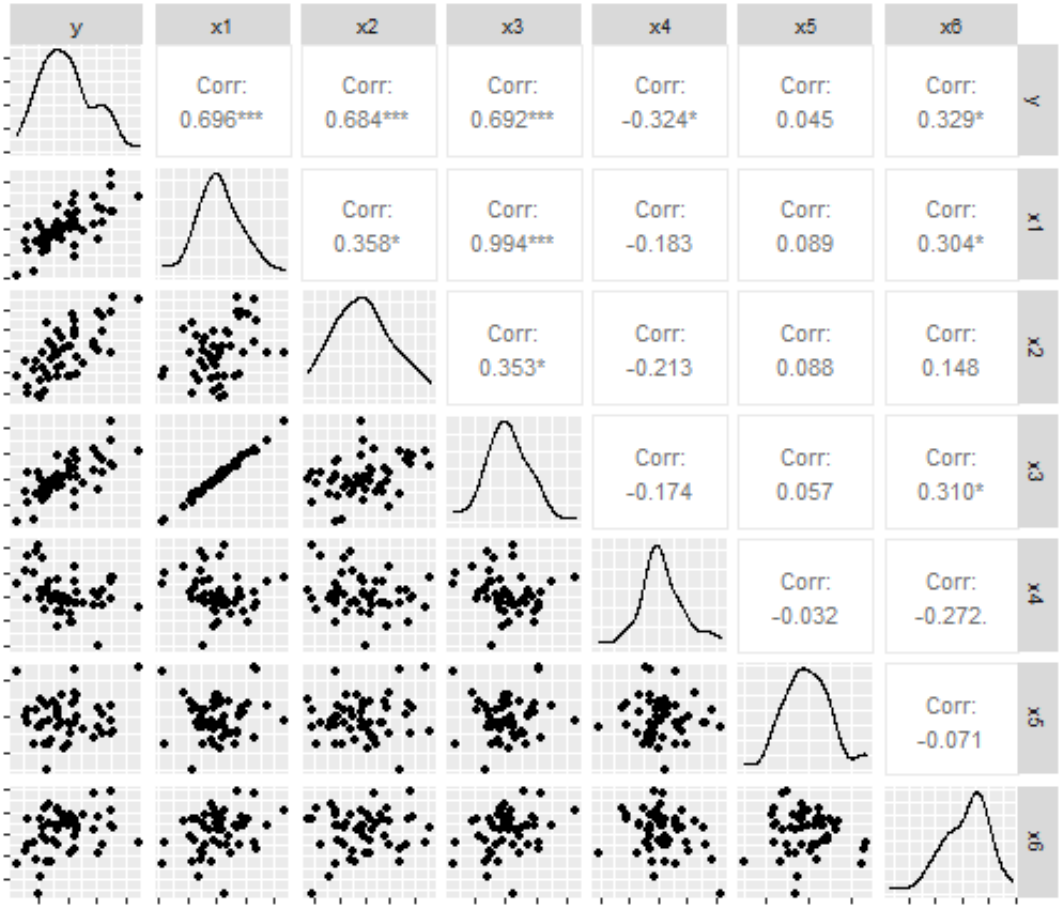
Se dispone de 6 variables regresoras:

- x_1 y x_2, que intervienen en el modelo y son independientes entre sí;
- x_3, que está muy correlada con x_1;
- x_4, x_5 y x_6, que son independientes entre sí y tampoco están relacionadas con el resto de variables regresoras o la variable respuesta.

Ejemplo

```
set.seed(100) # para reproducir los resultados
n <- 50 # tamaño muestral
x1 <- rnorm(n, sd=1)
x2 <- rnorm(n, sd=1)
x3 <- 0.9*x1 + rnorm(n, sd=0.1)  # muy correlada con x1
x4 <- rnorm(n, sd=0.5)
x5 <- rnorm(n, sd=0.5)
x6 <- rnorm(n, sd=0.5)
y <- x1 + x2 + rnorm(n, sd=1)

datos <- data.frame(y=y, x1=x1, x2=x2, x3=x3, x4=x4, x5=x
ggpairs(datos) +
  theme(axis.text=element_blank())
```

Ejemplo


y
x1
x2
x3
x4
x5
x6
Corr: 0.696***
Corr: 0.684***
Corr: 0.692***
Corr: -0.324*
Corr: 0.045
Corr: 0.329*
Corr: 0.358*
Corr: 0.994***
Corr: -0.183
Corr: 0.089
Corr: 0.304*
Corr: 0.353*
Corr: -0.213
Corr: 0.088
Corr: 0.148
Corr: -0.174
Corr: 0.057
Corr: 0.310*
Corr: -0.032
Corr: -0.272.
Corr: -0.071

Ejemplo: modelo con todas las variables

```
modelo_completo <- lm(y ~ ., data=datos)
summary(modelo_completo)
```

```
##
## Call:
## lm(formula = y ~ ., data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25219 -0.44265 -0.09985  0.61888  2.17817
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.09934    0.15144  -0.656    0.515
## x1           0.80947    1.75254   0.462    0.646
## x2           0.72951    0.13268   5.498 1.95e-06 ***
## x3           0.29072    1.93098   0.151    0.881
## x4          -0.52830    0.39122  -1.350    0.184
## x5          -0.11049    0.26702  -0.414    0.681
## x6           0.30636    0.34625   0.885    0.381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.019 on 43 degrees of freedom
## Multiple R-squared:  0.7242,    Adjusted R-squared:  0.6858
## F-statistic: 18.82 on 6 and 43 DF,  p-value: 1.343e-10
```

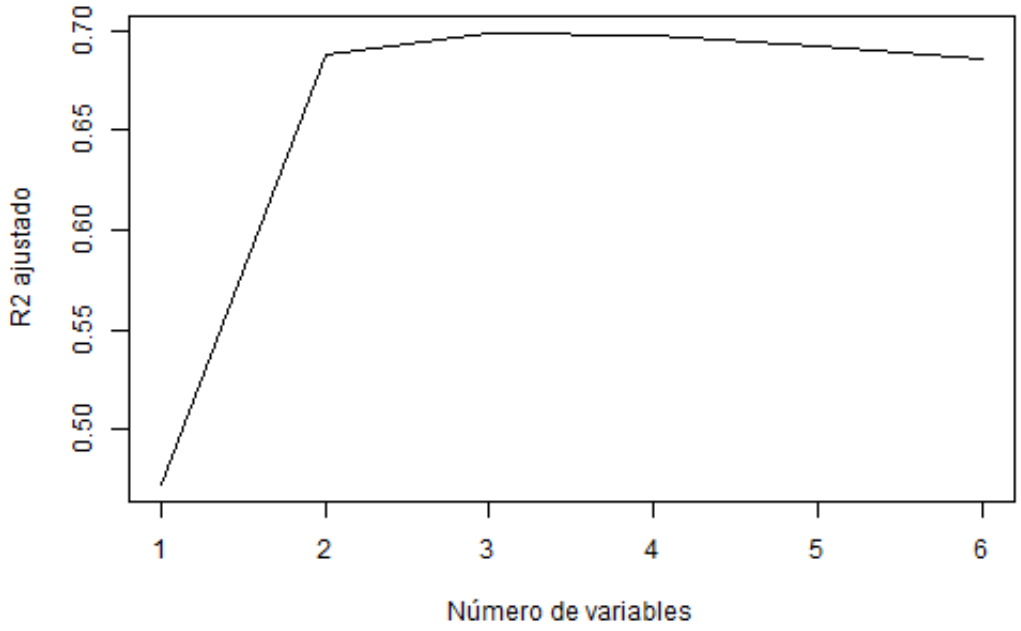
Ejemplo: método exhaustivo

```
modelo_todos <- leaps::regsubsets(y ~ ., data=datos)
resumen_todos <- summary(modelo_todos)
resumen_todos
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = datos)
## 6 Variables  (and intercept)
##    Forced in Forced out
## x1     FALSE      FALSE
## x2     FALSE      FALSE
## x3     FALSE      FALSE
## x4     FALSE      FALSE
## x5     FALSE      FALSE
## x6     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4  x5  x6
## 1  ( 1 ) "*" " " " " " " " " " "
## 2  ( 1 ) "*" "*" " " " " " " " "
## 3  ( 1 ) " " "*" "*" "*" " " " "
## 4  ( 1 ) "*" "*" " " "*" " " "*"
## 5  ( 1 ) "*" "*" " " "*" "*" "*"
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

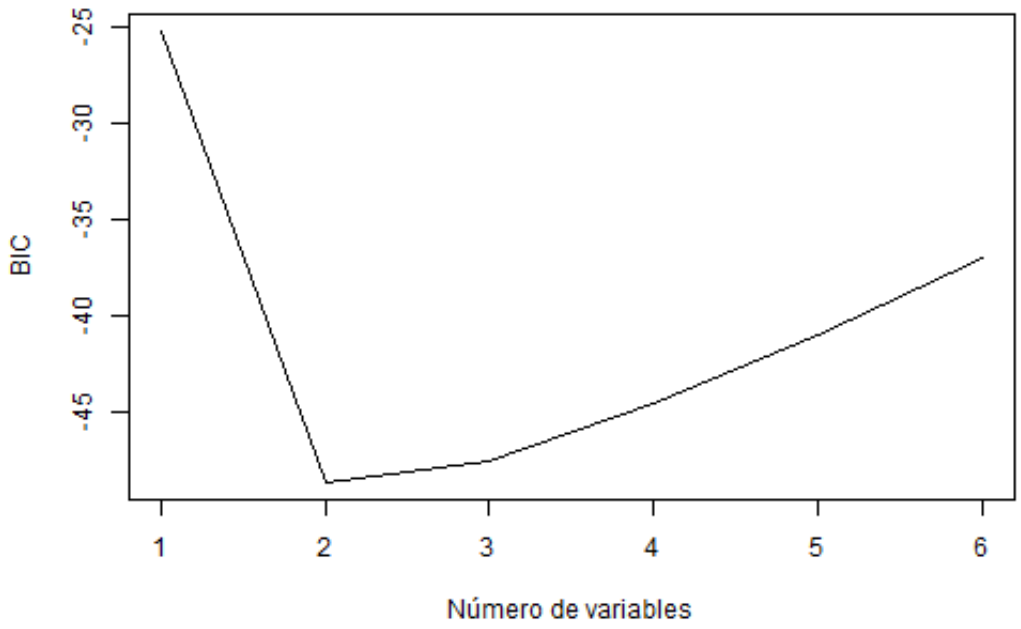
Ejemplo: comparación de modelos

```
plot(resumen_todos$adjr2 ,xlab =" Número de variables ",
     ylab="R2 ajustado",type="l")
```



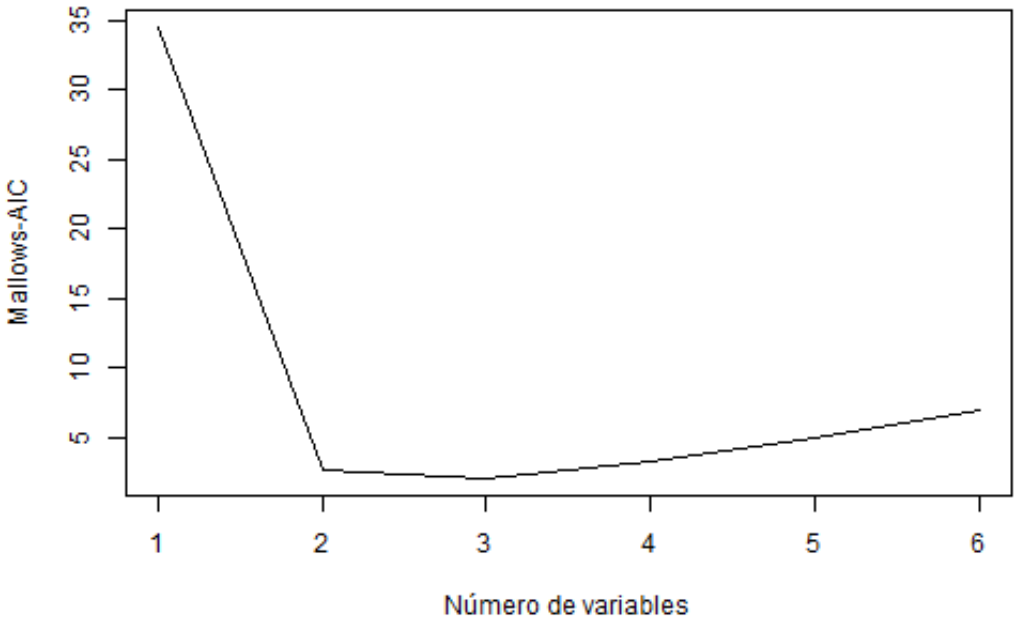
Ejemplo: comparación de modelos

```
plot(resumen_todos$bic ,xlab =" Número de variables ",
     ylab="BIC",type="l")
```



Ejemplo: comparación de modelos

```
plot(resumen_todos$cp ,xlab =" Número de variables ",
     ylab="Mallows-AIC",type="l")
```



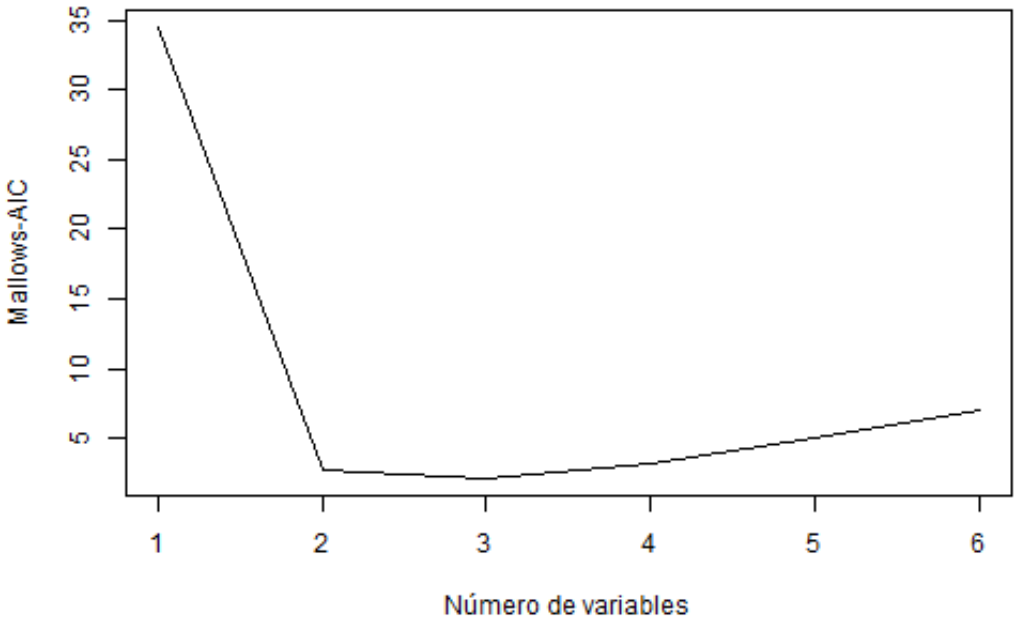
Ejemplo: paso a paso

```
modelo_forward <- regsubsets(y ~ ., data=datos, method='f
resumen_forward <- summary(modelo_forward)
resumen_forward
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = datos, method = "forward'
## 6 Variables  (and intercept)
##    Forced in Forced out
## x1     FALSE      FALSE
## x2     FALSE      FALSE
## x3     FALSE      FALSE
## x4     FALSE      FALSE
## x5     FALSE      FALSE
## x6     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: forward
##          x1  x2  x3  x4  x5  x6
## 1  ( 1 ) "*" " " " " " " " " " "
## 2  ( 1 ) "*" "*" " " " " " " " "
## 3  ( 1 ) "*" "*" " " "*" " " " "
## 4  ( 1 ) "*" "*" " " "*" " " "*"
## 5  ( 1 ) "*" "*" " " "*" "*" "*"
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

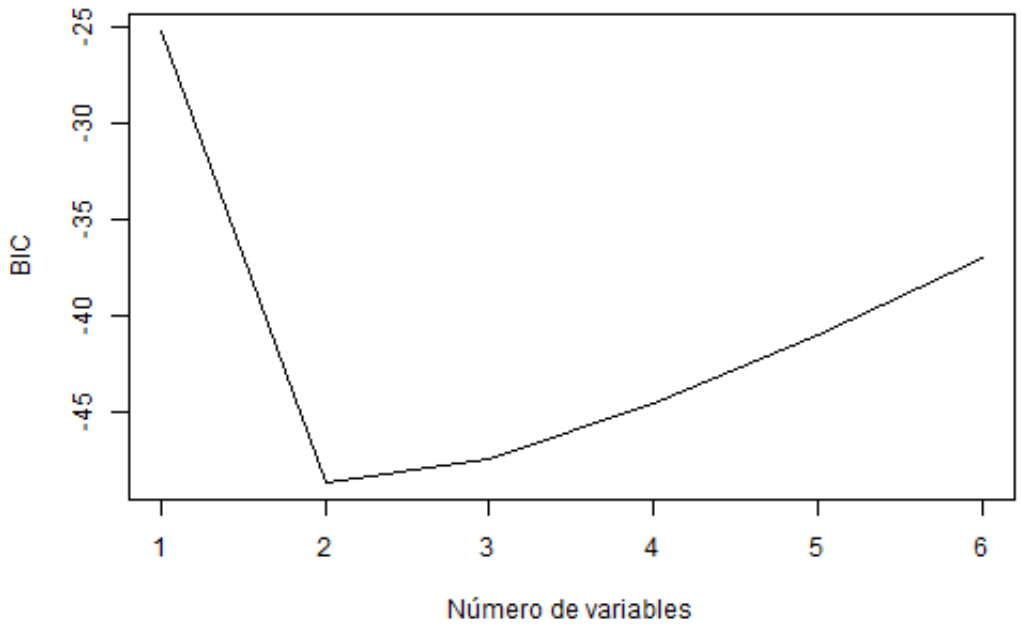
Ejemplo: comparación de modelos

```
plot(resumen_forward$cp ,xlab =" Número de variables ",
     ylab="Mallows-AIC",type="l")
```



Ejemplo: comparación de modelos

```
plot(resumen_forward$bic ,xlab =" Número de variables ",
     ylab="BIC",type="l")
```



Reducción de la dimensión: componentes principales

El objetivo es pasar de datos d-dimensionales a datos p-dimensionales, con $p < d$, sin perder mucha información. Para ello proyectamos en p direcciones adecuadas.

En componentes principales, la información se supone relacionada directamente con la varianza de las proyecciones.

$X = (X_1, \ldots, X_d)$ tiene matriz de covarianzas Σ con autovalores $\lambda_1 > \lambda_2 > \cdots > \lambda_d > 0$.

La proyección (signada) de X en la dirección a con $||a||_2 = 1$, es

$$||x||_2 \cos(\alpha) = a'X = \sum_{j=1}^{d} a_j X_j,$$

donde α es el ángulo entre X y a.

La varianza de la proyección es $\mathrm{Var}(a'X) = a'\Sigma a$.

La primera componente principal

La dirección de la primera componente principal resuelve:

$$\max a'\Sigma a \quad \text{s.a.} \quad a'a = 1$$

Diagonalizamos $\Sigma = V\Lambda V'$. Se tiene que:

$$a'\Sigma a = a'V\Lambda V'a = w'\Lambda w = \sum_{i=1}^{d} \lambda_i w_i^2 \leq \lambda_1$$

ya que $w = (w_1, \ldots, w_p)' = V'a$ verifica $\sum_{i=1}^{d} w_i^2 = 1$ para todo a con $\|a\|_2 = 1$.

Además, $v_1'\Sigma v_1 = \lambda_1 v_1'v_1 = \lambda_1$

La conclusión es que **la dirección de la primera componente principal, que denotamos v_1, es la definida por un autovalor normalizado de λ_1. Además, λ_1 es la varianza de las proyecciones.**

Esta solución es única salvo por el signo.

El resto de componentes

La dirección de la segunda componente principal resuelve:

$$\max a'\Sigma a \quad \text{s.a.} \quad a'a = 1, \quad a'v_1 = 0.$$

La segunda restricción impone que las dos primeras componentes sean incorreladas de manera que no contengan información redundante.

Puede demostrarse (ejercicio) que esta segunda componente viene dada por el autovector normalizado del segundo autovalor.

Análogamente se definen las d componentes principales v_j, $j = 1, \ldots, d$ y se deduce que **son los d autovectores normalizados de Σ correspondientes a los autovalores ordenados de mayor a menor.**

Componentes principales muestrales

Sea ahora $X = (x_1, \ldots, x_n)'$ la matriz $n \times d$ de datos **centrados**.

La matriz de covarianzas muestral $S = n^{-1} X'X$.

Si diagonalizamos S, tenemos $S = V \Lambda V'$, donde

- $V = (v_1, \ldots, v_d)$ es la matriz ortonormal cuyas columnas son los autovectores normalizados (luego $V'V = I_d$)
- $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_d\}$ es la matriz diagonal formada por los autovalores en orden decreciente.

Las proyecciones de los datos en las direcciones de las componentes principales son $W = XV$.

Reduciremos la dimensión de d a p si retenemos solo las p primeras:

$$\tilde{X} = XV_p, \quad V_p = (v_1, \ldots, v_p), \quad (V_p'V_p = I_p).$$

Componentes principales muestrales

Otra forma de escribir la diagonalización (dado que $V\Lambda = (\lambda_1 v_1, \ldots, \lambda_p v_p)$)

$$S = \sum_{j=1}^{d} \lambda_j v_j v_j'.$$

Si hay $p < d$ autovalores grandes y el resto aproximadamente cero, la estructura de covarianzas de los datos se puede aproximar a partir de las p primeras componentes principales.

¿Cuántas componentes principales se deben retener?

Proporción de varianza explicada por las p primeras componentes:

$$V_p = \frac{\lambda_1 + \cdots + \lambda_p}{\lambda_1 + \cdots + \lambda_d}.$$

Se suelen representar los valores (p, V_p).

Ejemplo

El fichero de datos cereal.csv contiene información nutricional de 77 marcas de cereales.

- *Name*: marca
- *mfr*: fabricante (A = American Home Food Products; G = General Mills; K = Kelloggs; N = Nabisco; P = Post; Q = Quaker Oats; R = Ralston Purina)
- *type*: frío (C) o caliente (H)
- *calories*: calorías por ración
- *protein*: proteínas (g)
- *fat*: grasas (g)
- *sodium*: sodio (mg)
- *fiber*: fibra (g)
- *carbo*: carbohidratos complejos (g)
- *sugars*: azúcares (g)
- *potass*: potasio (mg)
- *vitamins*: vitaminas y minerales - 0, 25, or 100, del porcentaje recomendado
- *shelf*: estantería (1, 2, or 3, desde el suelo)
- *weight*: peso en onzas de una ración
- *cups*: número de tazas de una ración
- *rating*: calificación (posiblemente en una encuesta de consumo)

Más información sobre estos datos en este enlace.

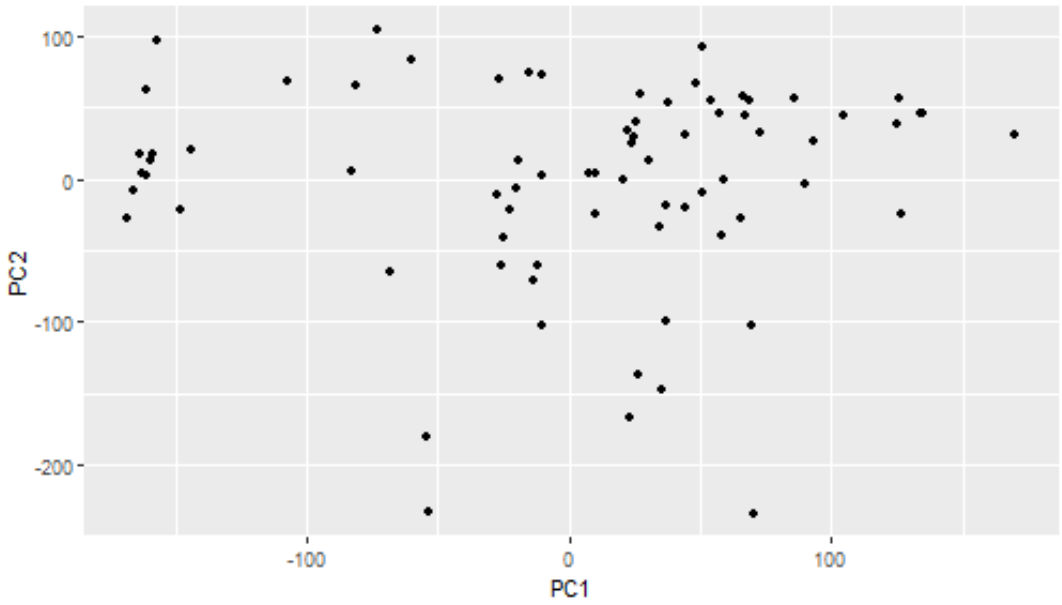
Ejemplo

```
cereal <- read.csv('https://verso.mat.uam.es/~joser.berre
knitr::kable(head(cereal[,1:10]), format = "markdown")
```

name	mfr	type	calories	protein	fat	sodium	fiber	c
100% Bran	N	C	70	4	1	130	10.0	
100% Natural Bran	Q	C	120	3	5	15	2.0	
All-Bran	K	C	70	4	1	260	9.0	
All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	
Almond Delight	R	C	110	2	2	200	1.0	
Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	

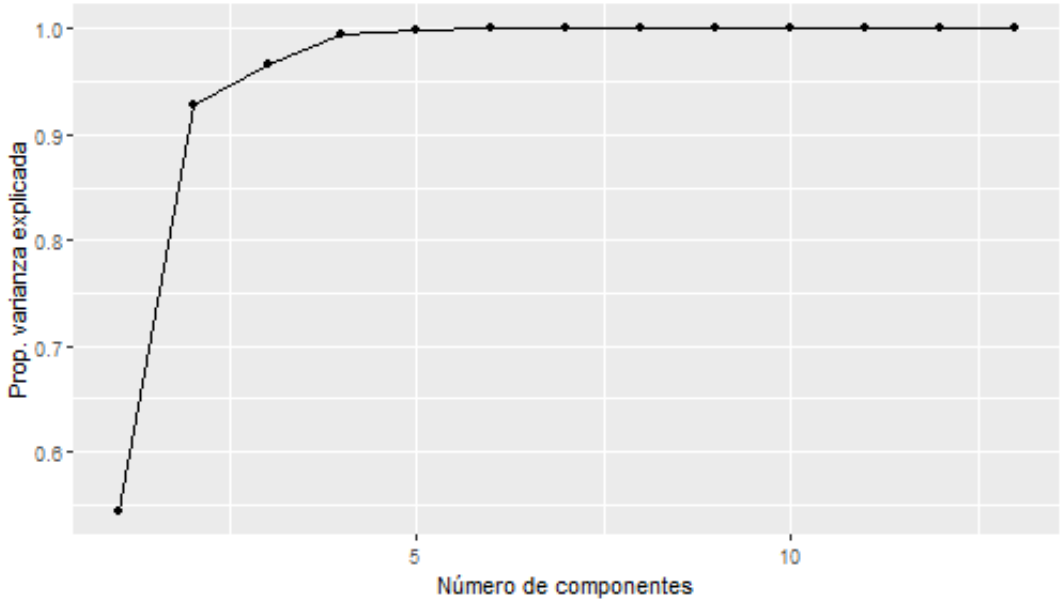
Ejemplo

```
prcomp_cereal <- prcomp(cereal[,-c(1, 2, 3)]) # quitamos
pca_cereal <- data.frame(
  PC1 = prcomp_cereal$x[, 1],
  PC2 = prcomp_cereal$x[, 2]
)

ggplot(pca_cereal, aes(x = PC1, y = PC2)) +
  geom_point()
```



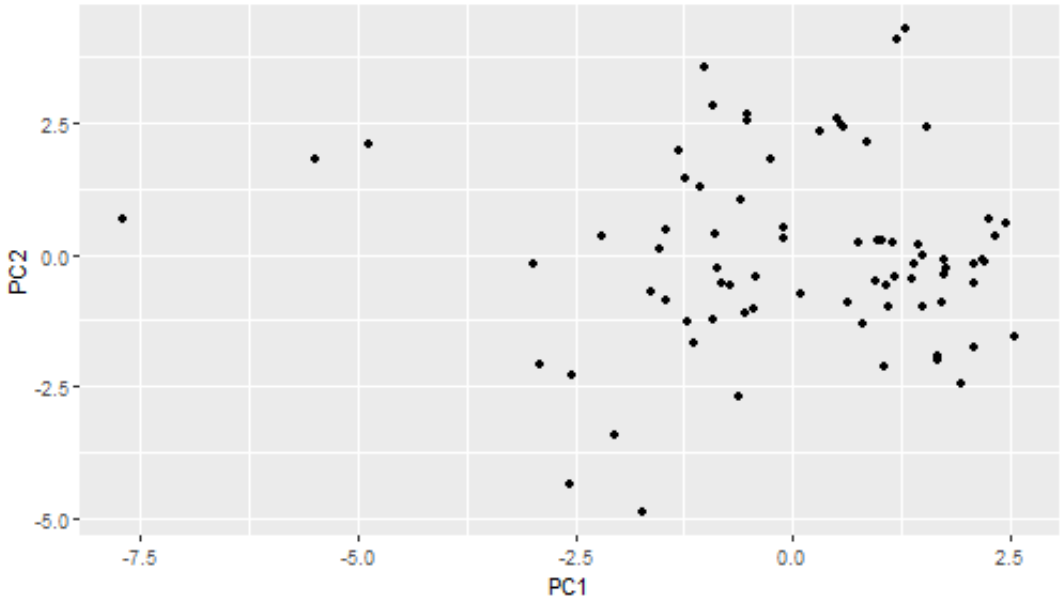
Ejemplo

```
var_acumulada <- cumsum(prcomp_cereal$sdev^2) / sum(prcom
sdev <- data.frame(indice = 1:13, var_acumulada)

ggplot(sdev, aes(x = indice, y = var_acumulada)) +
  geom_line() +
  geom_point() +
  labs(x = "Número de componentes", y = "Prop. varianza e
```



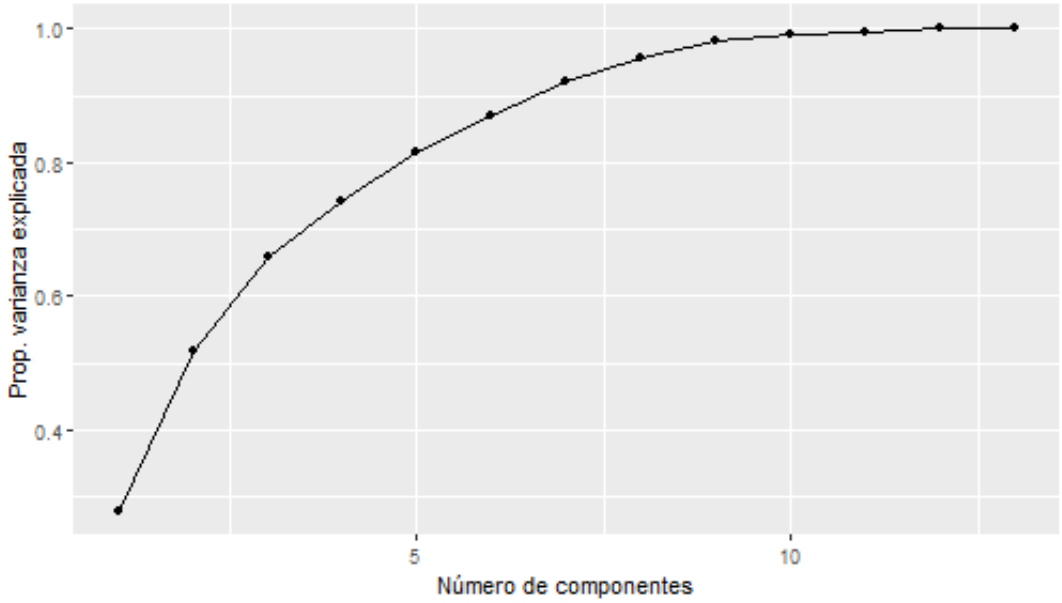
Ejemplo (variables estandarizadas)

```
prcomp_cereal <- prcomp(cereal[,-c(1, 2, 3)], scale. = TR
pca_cereal <- data.frame(
  PC1 = prcomp_cereal$x[, 1],
  PC2 = prcomp_cereal$x[, 2]
)

ggplot(pca_cereal, aes(x = PC1, y = PC2)) +
  geom_point()
```



Ejemplo

```
var_acumulada <- cumsum(prcomp_cereal$sdev^2) / sum(prcom
sdev <- data.frame(indice = 1:13, var_acumulada)

ggplot(sdev, aes(x = indice, y = var_acumulada)) +
  geom_line() +
  geom_point() +
  labs(x = "Número de componentes", y = "Prop. varianza e
```



Descomposición en valores singulares

Las componentes principales estandarizadas son $\tilde{U} = W\Lambda^{-1/2}$.

$$\frac{1}{n}\tilde{U}'\tilde{U} = \frac{1}{n}\Lambda^{-1/2}W'W\Lambda^{-1/2} = \Lambda^{-1/2}V'\left(\frac{1}{n}X'X\right)V\Lambda^{-1/2}$$

Se cumple $\tilde{U}\Lambda^{1/2} = W = XV$, de donde se deduce la **descomposición en valores singulares de X**:

$$X = \tilde{U}\Lambda^{1/2}V' = UDV',$$

donde $U = n^{-1/2}\tilde{U}$ y $D = n^{1/2}\Lambda^{1/2}$.

- La matriz $\tilde{U}$ está formada por las componentes principales de los datos estandarizadas. Por tanto, $U'U = I_d$
- La matriz diagonal $D = \sqrt{n}\Lambda^{1/2}$ está formada por $\sigma_i := \sqrt{n\lambda_i}$.
- La matriz ortonormal V está formada por las direcciones de las componentes principales en columnas.

Descomposición en valores singulares

Forma equivalente de escribir la matriz de datos

$$X = \sum_{j=1}^{d} \sigma_j u_j v_j'.$$

Si hay $p < d$ autovalores grandes y el resto son pequeños,

$$X \approx \sum_{j=1}^{p} \sigma_j u_j v_j' = \tilde{X}.$$

No perdemos mucho por considerar las p primeras componentes.

Mínimos cuadrados y componentes

Expresión alternativa de mínimos cuadrados:

$$\hat{\beta} = (X'X)^{-1}X'Y = VD^{-1}U'Y = \sum_{j=1}^{d} \frac{u_j' y}{\sigma_j} v_j$$

Matriz de covarianzas de $\hat{\beta}$:

$$\mathrm{Var}(\hat{\beta}) = \sigma^2 (X'X)^{-1} = \sigma^2 \sum_{j=1}^{d} \frac{v_j v_j'}{\sigma_j^2}.$$

Las últimas componentes principales tienen bastante importancia en la matriz de covarianzas del estimador de mínimos cuadrados.

Regresión de componentes principales

Consiste en usar como matriz de datos $\tilde{X}$ en lugar de X.

Sumamos hasta p en lugar de hasta d en las expresiones anteriores.

La regresión de componentes principales asume implícitamente que las últimas componentes no son importantes a la hora de explicar la respuesta.

Otros métodos de reducción de la dimensión, como *mínimos cuadrados parciales (PLS, partial least squares)*, toman en cuenta la variable respuesta para construir las direcciones de proyección de las variables.

Regularización: ridge

Regularización: ridge

Hoerl y Kennard (1970) propusieron modificar la expresión del estimador de mínimos cuadrados añdiendo una constante $\lambda > 0$ a la diagonal de $X'X$

$$\hat{\beta}(\lambda) = (X'X + \lambda I)^{-1}X'Y = M(\lambda)\hat{\beta},$$

donde $\hat{\beta}$ es el estimador de mínimos cuadrados y $M(\lambda) = (X'X + \lambda I_p)^{-1}X'X$.

El estimador ridge resuelve

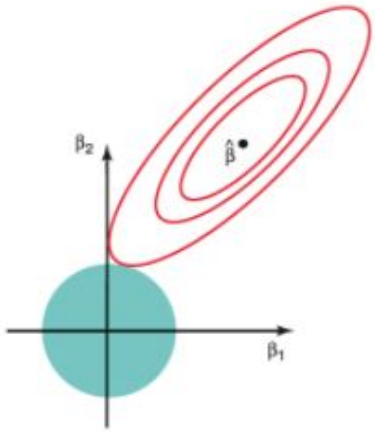
$$\hat{\beta}(\lambda) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||_2^2 + \lambda ||\beta||_2^2.$$

También resuelve el siguiente problema para un valor $c > 0$ adecuado:

$$\hat{\beta}(\lambda_c) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||_2^2 \quad \text{s.a.} \quad ||\beta||_2^2 \leq c$$

Ridge

El método ridge equivale a calcular el estimador de mínimos cuadrados restringido a una bola definida por la norma euclídea.



Ridge y descomposición en valores singulares

Recordamos $X = UDV'$. Dado que $(X'X + \lambda I)^{-1} = V(D^2 + \lambda I)^{-1}V'$,

$$\hat{\beta}(\lambda) = \sum_{j=1}^{d} \frac{\sigma_j^2}{\sigma_j^2 + \lambda} \frac{1}{\sigma_j}(u_j' y) v_j$$

Mínimos cuadrados

$$\hat{\beta} = \sum_{j=1}^{d} \frac{1}{\sigma_j}(u_j' y) v_j$$

Componentes principales

$$\hat{\beta}_{cp} = \sum_{j=1}^{p} \frac{1}{\sigma_j}(u_j' y) v_j$$

Regresores ortogonales

Si $X'X = I_d$, entonces $\hat{\beta}(\lambda) = (1+\lambda)^{-1}\hat{\beta}$.

$$\text{Sesgo}(\hat{\beta}(\lambda)) = -\frac{\lambda}{1+\lambda}\beta,$$

y

$$\text{Var}(\hat{\beta}(\lambda)) = \frac{1}{(1+\lambda)^2}\text{Var}(\hat{\beta}) = \frac{\sigma^2}{(1+\lambda)^2}I_d.$$

Si error cuadrático medio de un **vector** es $\text{ECM}(\hat{\theta}) = \text{E}(\|\hat{\theta} - \theta\|_2^2)$,

$$\text{ECM}(\hat{\beta}) = \frac{\lambda^2}{(1+\lambda)^2}\beta'\beta + \frac{\sigma^2 d}{(1+\lambda)^2}.$$

Esta expresión se minimiza para $\lambda^* = d\sigma^2/\beta'\beta$.

Theobald (1974): siempre existe $\lambda > 0$ tal que $\text{ECM}(\hat{\beta}(\lambda)) < \text{ECM}(\hat{\beta})$.

Selección del parámetro de regularización

Es necesario seleccionar un valor adecuado para el parámetro de regularización λ

Algoritmo de validación cruzada *leave-one-out*.

- Para cada uno de los posibles valores de λ y para cada $i = 1, \ldots, n$, calculamos $\hat{\beta}_{(-i)}(\lambda)$ usando todas las observaciones menos la i-ésima.
- Evaluamos λ mediante

$$\phi(\lambda) = \sum_{i=1}^{n} [Y_i - \hat{\beta}_{(-i)}(\lambda)' x_i]^2 = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i(\lambda)}{1 - h_{ii}(\lambda)} \right)^2,$$

donde $h_{ii}(\lambda)$ es el elemento i de la diagonal de $X(X'X + \lambda I)^{-1} X'$

- Seleccionamos $\hat{\lambda} = \arg\min \phi(\lambda)$.

Regularización: Lasso

Tiene una definición parecida al anterior pero sus propiedades son bastante diferentes.

El estimador lasso resuelve

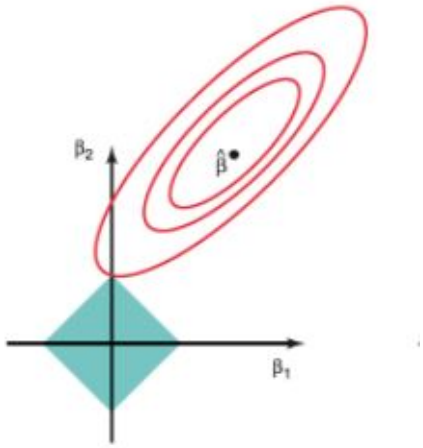
$$\hat{\beta}(\lambda) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||_2^2 + \lambda ||\beta||_1$$

También resuelve el siguiente problema para un valor $c > 0$ adecuado:

$$\hat{\beta}(\lambda_c) = \arg\min_{\beta \in \mathbb{R}^p} ||Y - X\beta||_2^2 \quad \text{s.a.} \quad ||\beta||_1 \leq c$$

Lasso

El método lasso equivale a calcular el estimador de mínimos cuadrados restringido a una bola definida por la norma L^1.

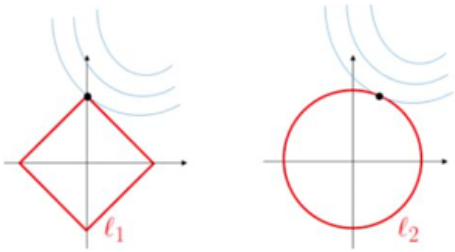


Lasso

La norma L^1 no es diferenciable: no es posible obtener expresiones explícitas de los estimadores lasso.

Problema de optimización convexo: computacionalmente factible

Principal ventaja de lasso frente a ridge: lleva a cabo automáticamente una selección de las variables. En el óptimo muchos de los coeficientes estimados son iguales a cero



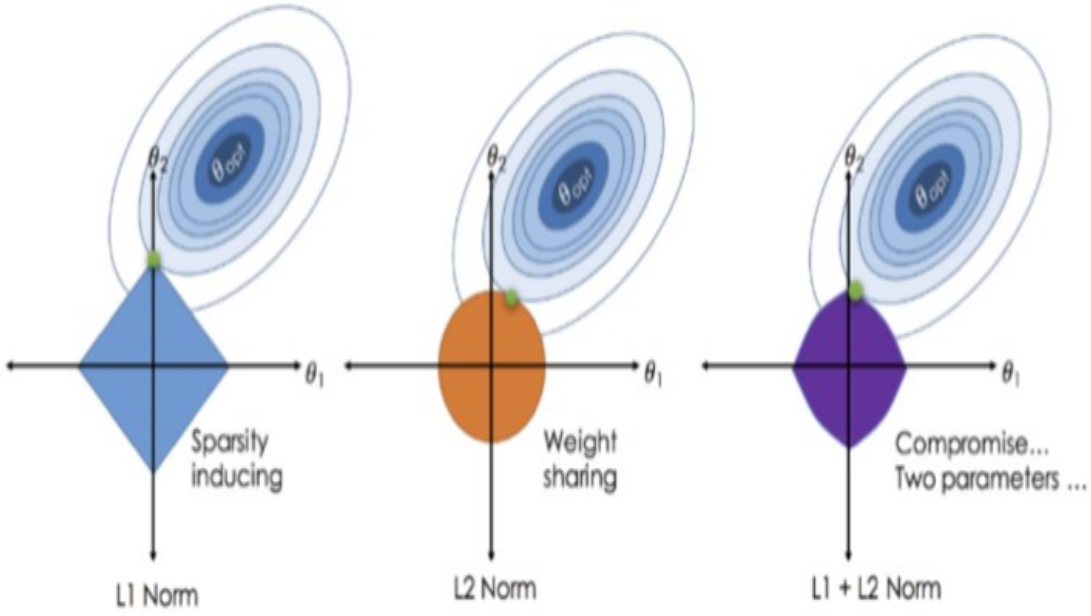
Ejemplo

Para ajustar el modelo mediante el método *ridge* y *lasso* usamos `glmnet`.

La forma de suministrar los datos cambia: el primer argumento es la matriz de variables regresoras y el segundo el vector con las observaciones de la variable respuesta.

El argumento `alpha=1` se utiliza para señalar que queremos aplicar *lasso*. Otros valores corresponden a diferentes métodos de regularización, como por ejemplo *ridge* o *elastic net* (un híbrido entre ridge y lasso que impone restricciones en las dos normas).

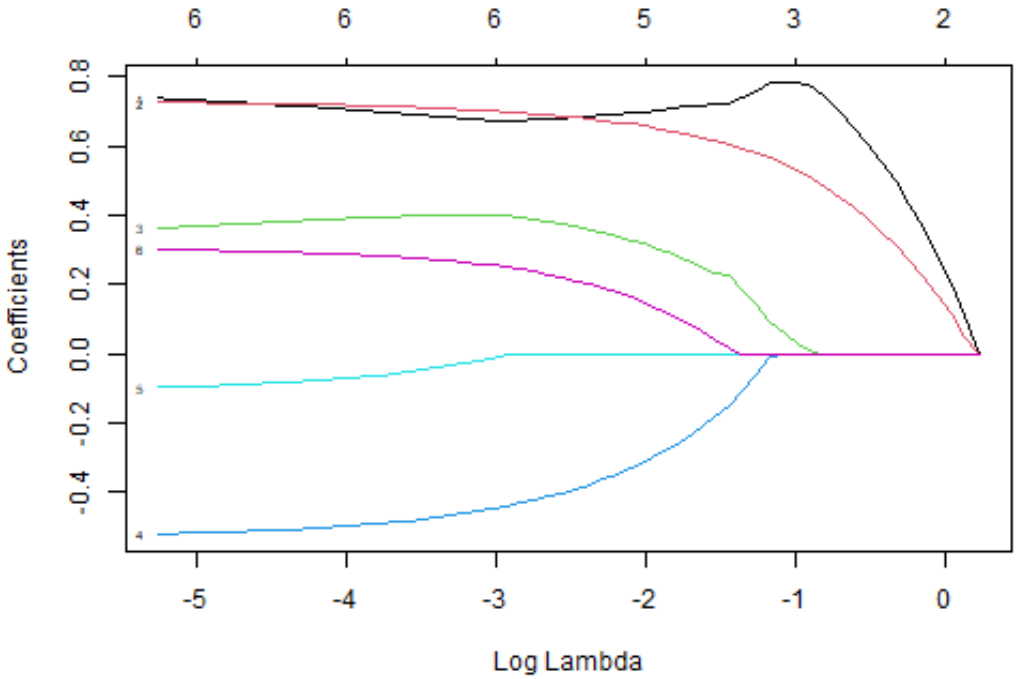
Ejemplo



Fuente del gráfico: towardsdatascience

Ejemplo

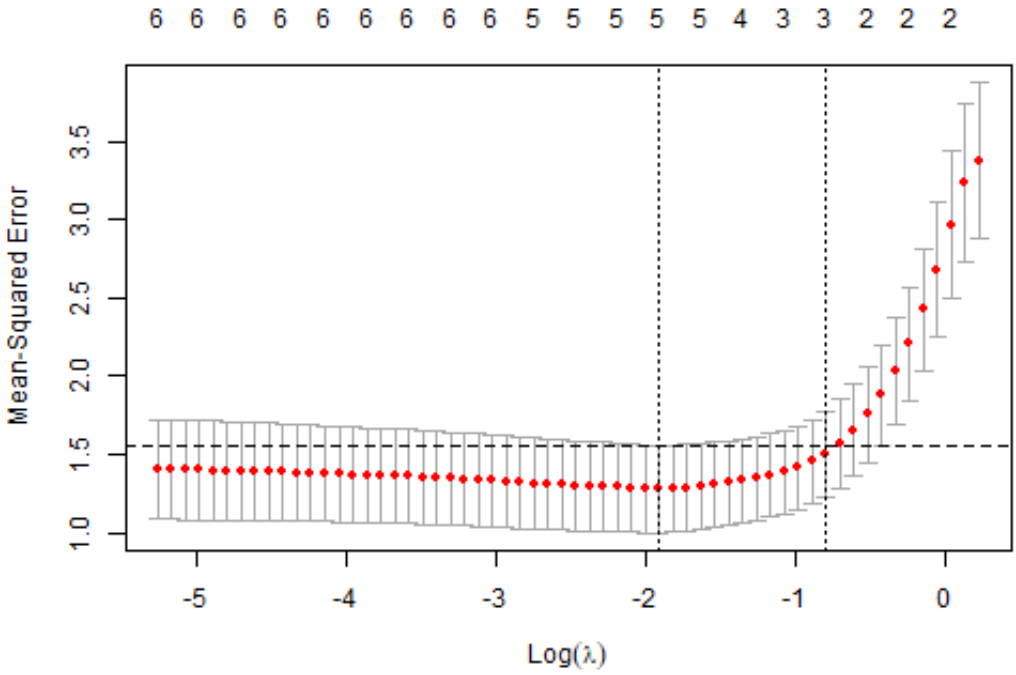
```
x <- as.matrix(datos[,-1])
y <- datos[,1]
modelo_lasso <- glmnet(x, y, alpha = 1)    # alpha = 1 (la
plot(modelo_lasso, xvar='lambda', label=TRUE)
```



Ejemplo

Usamos validación cruzada para elegir el valor de λ óptimo

```
lasso_cv <- cv.glmnet (x , y, alpha = 1)
plot(lasso_cv)
indice <- which(min(lasso_cv$cvm) == lasso_cv$cvm)
abline(h = lasso_cv$cvm[indice] + lasso_cv$cvsd[indice],
```

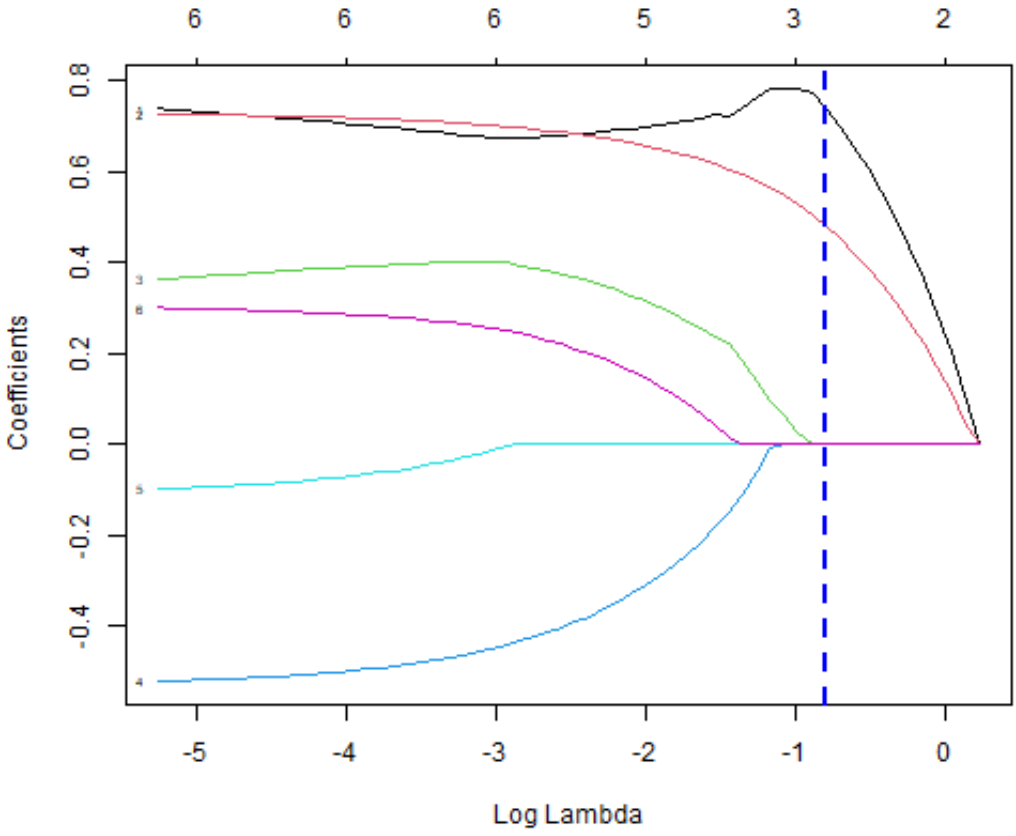


Ejemplo

- Los puntos rojos corresponden al error cuadrático medio (ECM) de predicción estimado mediante validación cruzada. Cada ECM se estima con un error típico.
- La línea vertical discontinua de la izquierda corresponde al valor de λ que da lugar al mínimo ECM estimado. Sea λ^* este valor, ECM^* el correspondiente ECM, y ET^* el error típico de ECM^*.
- La línea vertical discontinua de la derecha es el máximo λ tal que su ECM estimado es inferior a $\mathrm{ECM}^* + \mathrm{ET}^*$ (véase línea horizontal discontinua).
- Este último valor está incluido en `lasso_cv$lambda.1se`. En nuestro caso, el modelo para este valor de λ resulta incluir únicamente las variables x_1, x_2 y x_3.

Ejemplo

```
lambda.lasso = lasso_cv$lambda.1se
plot(modelo_lasso, xvar='lambda', label=TRUE)
abline(v = log(lambda.lasso), lty=2, col='blue', lwd=2)
```



Ejemplo

Se utiliza el argumento `lambda` para fijar un valor del parámetro de regularización):

```
lambda.lasso = lasso_cv$lambda.1se
modelo_final_lasso <- glmnet(x, y, alpha = 1, lambda = la
coef(modelo_final_lasso)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -0.1047409372
## x1           0.7385749527
## x2           0.4796344636
## x3           0.0001835796
## x4           .
## x5           .
## x6           .
```